Proceedings of the 2006 IEEE International Conference on Robotics and Automation
Orlando, Florida - May 2006

Maintaining Wireless Connectivity Constraints for Swarms in the Presence of Obstacles

Joel M. Esposito
US Naval Academy, WSE Dept.
Annapolis, MD
Email: esposito@usna.edu

Thomas W. Dunbar
Naval Postgraduate School
Monterey, CA
Email: twdunbar@gmail.com

Abstract— **The low power requirements of many small radio modems suggest that robust operation is best attained when the transmitter/receiever pair is: (1) separated by less than some maximum distance (Range); and (2) not obstructed by large dense objects (Line-of-Sight). Therefore to maintain a wireless link between two robots, it is desirable to comply with these two spatial constraints. Given a swarm of point robots with specified initial and final configurations and a set of desired communication links consistent with the above criteria, we explore the problem of designing inputs to achieve the final configuration while preserving the desired links for the duration of the motion. Some interesting conclusions about the feasibility of the problem are offered. An algorithm is provided and its operation is demonstrated through both simulation and experimentation on Koala Robots.**

I. Introduction

Recently much attention has been given to maintaining network connectivity in swarms of mobile robots equipped with low power wireless transmitter/receivers. Power limitations dictate constraints on the relative distance between two communicating robots within the swarm (referred to here as *Range Constraints*). Many other groups have developed control frameworks for maintaining swarm connectivity under range constraints – often in obstacle free environments. We discuss some of these works below. However considerably less attention has been given to *Line-of-Sight Constraints* – necessitated by difficulty of reliably transmitting wireless messages through large dense obstacles. In addition to these constraints the robots may have some overall motion objectives (either individually or as a group).

In this paper we address the problem of navigating the swarm to a specific final configuration, while maintaining a pre-specified list of wireless links between robots (Range plus Line-of-Sight). After a review of related work below, we provide a formal problem statement in Sect. II. We consider existence of solutions in Sect. III and discuss a key attribute of any acceptable solution. In Sect. IV we introduce potential fields for Goal, Range and Line of Sight objectives. Sect. V discusses how to compose these sometimes disparate objectives and provides a computational algorithm for assigning motion directions. Simulation and hardware-based experimental demonstrations of the algorithm's operation are included in Sect. VI followed by concluding remarks in Sect. VII.

Much work has been done in the area of swarms and flocking recently, we only mention the most closely related work in an effort to distinguish the class of problems we consider. Due to space limitations this discussion is by no means comprehensive. We loosely categorize work in this area according to the following identifiers: 1. fixed/variable relative pose; 2. centralized/decentralize control; 3. fixed/dynamic connection topology; 4. individual/swarm-wide objective(s); and 5. obstacle/obstacle-free workspaces. For example, the problem of *Formation control* is considered in [1]. The basic problem involves fixed relative pose among the swarm members, centralized control (*i.e.*, leader), fixed connection topology, swarm-wide objective (leader position) and obstacles in the workspace. In contrast [4], [5], [9] consider the *Flocking Problem*. No relative pose is prescribed among the the swarm members and no leader is prescribed. Instead a range constraint and velocity matching controller result in a collective behavior. The flock converges to some steady state velocity, although each member does not persue an individual objective. Fixed and dynamic connectivity are considered but no obstacles are present. In [8] the flocking problem is also considered in a slightly different framework. A swarm wide objective is supplied and some obstacles are considered, connectivity is dynamic. Our work is characterized as follows:

- unlike Formation Control, no relative pose between robots is specified and no lead robot is designated;
- we do not explicitly address dynamic connection topology in this paper;
- unlike Flocking, each swarm member has a distinct goal position rather than a swarm wide desired velocity;
- we consider obstacles avoidance, but unlike the work considered above, we also account for their affect on connectivity via the line-of-sight constraint.

Perhaps the most closely related work is [2] in which a geometric connectivity constraint (range), an initial and final position, and a desired communication graph is specified. However, obstacles prevented the presented solution method-

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **MAY 2006**	2. REPORT TYPE	3. DATES COVERED **00-00-2006 to 00-00-2006**
4. TITLE AND SUBTITLE **Maintaining Wireless Connectivity Constraints for Swarms in the Presence of Obstacles**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **United States Naval Academy,Department of Weapons and Systems Engineering,Annapolis,MD,21402**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR’S ACRONYM(S)
		11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The low power requirements of many small radio modems suggest that robust operation is best attained when the transmitter/receiever pair is: (1) separated by less than some maximum distance (Range); and (2) not obstructed by large dense objects (Line-of-Sight). Therefore to maintain a wireless link between two robots, it is desirable to comply with these two spatial constraints. Given a swarm of point robots with specified initial and final configurations and a set of desired communication links consistent with the above criteria we explore the problem of designing inputs to achieve the final configuration while preserving the desired links for the duration of the motion. Some interesting conclusions about the feasibility of the problem are offered. An algorithm is provided and its operation is demonstrated through both simulation and experimentation on Koala Robots.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**6**	

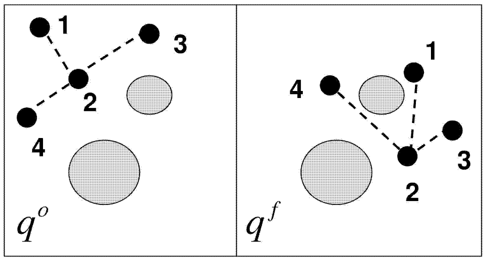

Fig. 1. The basic problem considered in this paper. Design a control law to guide the swarm from q^o to q^f while maintaining desired communication links (Range plus Line-of-Sight).

ology from being applied. In addition they consider multi-hop network connections.

II. Problem Statement

Given n point robots, let $q_i \in \mathbb{R}^2$ be the state vector of robot i. The robots operate in a subset of the plane $C \subset \mathbb{R}^2$ which is populated with obstacles defined by compact sets O_j, $j = 1, \dots, m$.

Motion is generated according to the dynamics

$$\dot{q}_i = u_i \tag{II.1}$$

where $u_i \in U \subset \mathbb{R}^2$; and q_i is only permitted to evolve in the free space $C_{free} = C - \bigcup_{j=1}^{m} O_j$. Occasionally we will use $q \in \mathbb{R}^{2n}$ to denote the swarm state – the concatenation of states $q_1 \dots q_n$; u to represent the concatenation of the input vectors; and $\dot{q} = u$ to represent the collective swarm dynamics.

Any given swarm state q induces a communication graph $G(q) = (V, E)$. Each vertex in the graph, $v_i \in V$ represents a robot and each edge $e_{ij} \in E$ represents a wireless communication link between robots i and j. The edge e_{ij} is added to the graph if both of the following conditions are met:

1) *Range*: $d(q_i, q_j) \leq \rho_{\max}$ where $\rho_{\max}$ is some positive constant indicating the maximum effective range of the transmitter; and
2) *Line-of-Sight*: $\exists x(\alpha) \in C_{free}$, $\forall \alpha \in [0,1]$, such that $x(\alpha) = \alpha q_i + (1-\alpha) q_j$,

note that d indicates distance as measured by the Euclidian metric. Both constraints model the power limitations of small wireless transmitters discussed in Sect. I. A configuration q is said to be *connected* if the induced communication graph G is connected (*i.e.* if for any node pair i, j there exists an edge path of arbitrary length between them).

We are concerned with the following problem (see Fig. 1), which requires the entire swarm to move to a desired position while maintaining certain communication links, G^*.

Problem 2.1: Given an initial connected configuration $q^o = q(t^o)$ and a final connected configuration q^f and a graph G^* such that $G(q^o) \supseteq G^*$ and $G(q^f) \supseteq G^*$, determine a function $\mathcal{U} : [t^o, t^f] \to U$ such that

1) $q(t^f) = q^f$ (*i.e., Goal* directed motion);
2) $G(q(t)) \supseteq G^*$, $\forall t \in [t^o, t^f]$ (*Line-of-Sight* and *Range*).

III. Existence of solutions

Clearly there are certain combinations of the free space C_{free} and the desired connectivity graph G^* for which the problem may not have a solution. Furthermore, even when a solution exists, there are certain classes of algorithms incapable of solving the problem. The concept of homotopy [6] is intimately related to these existence questions.

A. Homotopy Definitions

If $\tau_1, \tau_2 : [t^o, t^f] \to C_{free}$ are continuous maps (paths), we say that τ_1 and τ_2 are *homotopic* if there exists a continuous map $T : [t^o, t^f] \times [0,1] \to C_{free}$ such that

$$T(t, 0) = \tau_1(t) \tag{III.1}$$
$$T(t, 1) = \tau_2(t)\ , \forall t. \tag{III.2}$$

If such a function exists, we say T is a homotopy. Note that we do not require the endpoints to remain fixed. This homotopy defines an equivalence relation on paths.

If $\tau(t^o) = \tau(t^f)$ the path is considered a loop λ. We can apply the homoptopy equivalence relation to loops as well. The trivial loop is the constant loop $\lambda(t) = \lambda(t^o)$.

Of particular interest in this paper is the *straight-line homotopy*

$$T(t, s) = (1 - s)\tau_1(t) + s\tau_2(t), \tag{III.3}$$

due to its obvious connection to the line of sight constraint. If two path τ_1, τ_2 have a straight line homotopy then the line of sight constraint is preserved for all t. If the range constraint is violated at any point on the trajectory, the straight-line homotopy can be used to correct the condition.

B. Intrinsic lack of solution

Naturally in the case when C_{free} is not path-connected, and q^0 and q^f lie in different connected components, there is no solution to the motion planning problem.

Furthermore, if C_{free} is multiply connected and G^* contains cycles, solutions do not exist for all choices of q^o or q^f. For a given cycle $G^c \subseteq G^*$, one can connect the points q_i^o corresponding to the vertices in the cycle to form a loop λ^o using straight-line segments; likewise a corresponding loop λ^f, using q^f, can be constructed using the same vertices. See Figure 2 for an example. If these two loop are not in the same homotopic equivalence class, it implies the loops wrap around the obstacles in such a way that is impossible to go from q^0 to q^f without disconnecting some edges, then no solution to the problem exists.

Remark 3.1: To ensure the existence of solutions in this paper we only consider path-connected free spaces; and q^0, q^f such that loops corresponding to any cycles of G^* are homotopic to the constant loop.

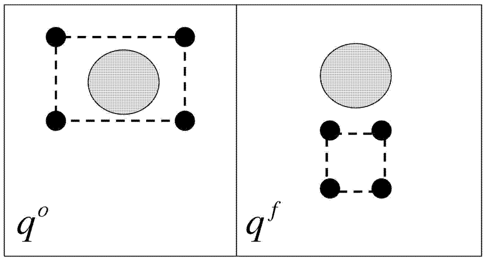

Fig. 2. Frames depict example of a combination of G^*, q^o and q^f which are not feasible. Loops in left and right frames are not in the same homotopic equivalence class.

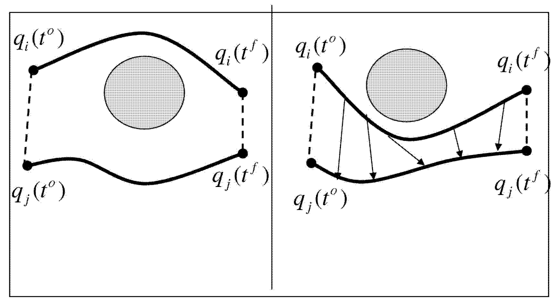

Fig. 3. The left frame illustrates a situation where Line-of-Sight is not maintained (no straight line homotopy between paths; loop is not homotopic to constant loop). The right frame shows two paths that maintain Line of Sight (straight line homotopy between paths; loop is homotopic to the constant loop).

C. Attribute of Complete Solution Algorithms

As remarked earlier, in order to maintain an edge e_{ij}, $\forall t \in [t^o, t^f]$, there must exist a straight-line homotopy between $q_i(t)$ and $q_j(t)$. Intuitively a necessary (not sufficient) condition for such a solution is that paths $q_i(t)$ and $q_j(t)$ must pass around the same "side" of an obstacle. See Figure 3. Therefore, if G^* is connected *all* robots, in some loose sense, must collectively pass around the same "side" of every obstacle– *i.e.* the swarm cannot "split".

Remark 3.2: An interesting interpretation of this is that a truly distributed controller is incapable of solving the problem for arbitrary initial conditions. Either some "lead" robot must select a path-class for the entire swarm or some type of bi-direction messaging must be used to reach a dynamic consensus on path-class selection.

This notion is difficult to formalize however. The traditional path-homotopy equivalence relation does not apply because the end points of $q_i(t)$ and $q_j(t)$ do not coincide. Also, general homotopy does not preserve the line-of-sight constraint; and the straight-line homotopy does not induce an equivalence relation (it lacks transitivity). Instead (see Figure 3) consider the loop resulting from $\lambda_{ij} = [q_i(t)] \cdot [q_i(t^f) \rightarrow q_j(t^f)] \cdot [q_j(t)]^{-1} \cdot [q_j(t^o) \rightarrow q_i(t^o)]$. Requiring the paths $q_i(t)$ and $q_j(t)$ to be on the same "side" of the obstacle, means this loop is homotopic to (in the same equivalence class as) to the constant loop.

IV. Potential functions

A. Review: Navigation Functions

In this paper we use Navigation Functions as the basis for ensuring the goal completion portion of the problem ($q \rightarrow q^f$). *Navigation Functions* are artificial potential fields that simultaneously provide obstacle avoidance and almost everywhere convergence to a goal configuration [7].

Definition 4.1: Navigation Function: For robot i, a scalar map $\phi_i^{goal} : C_{free} \rightarrow [0,1]$ is a *Navigation Function* if it is:

1) polar at q_i^f (i.e., has a unique minimum on the path connected component of C_{free} containing q_i^f);
2) admissible on C_{free} (i.e. it is uniformly maximal on the boundary of C_{free});
3) a Morse function (i.e. its Hessian is nonsingular at the critical points);
4) smooth on C_{free} (i.e. at least C^2).

As an example consider that , in the simplest case of circular obstacles in a circular workspace, a navigation function for robot i can be defined as:

$$\phi_i^{goal}(q_i) = \frac{d^2(q_i, q_i^f)}{[d^k(q_i, q_i^f) + \prod_{j=0}^{M} d(q_i, O_j)]^{1/k}}. \qquad \text{(IV.1)}$$

Where O_j is obstacle j, O_0 is the boundary of the workspace and the parameter k must be selected high enough that all local minima, except at q_i^f, disappear. The selection of k makes it difficult to compute navigation function for dynamic work spaces.

Remark 4.2: It is known that workplaces with M obstacles will inevitably possess M saddle points. Note that emerging from each saddle point, is a stable manifold connecting the saddle to other extrema. Initial positions on different sides of these manifolds evolve in different path classes around the obstacle associated with the saddle point. Therefore, if $e_{ij} \in G(q^0)$ but the line segment connecting $q^i(t^0)$ and $q^j(t^0)$ crosses the stable manifold, the line of sight constraint between i and j will not be maintained.

B. Range

The range constraint dictates that if $e_{ij} \in G^*$ then $d(q_i, q_j) \leq \rho_{\max}$. This is enforced by a potential

$$\phi_{ij}^{range}(q_i, q_j) = \begin{cases} 0 & d(q_i, q_j) < \rho_{\max} \\ d^2(q_i, q_j) - \rho_{\max}^2 & d(q_i, q_j) \geq \rho_{\max}. \end{cases}$$

Note the potential only posses minima at configurations where the range constraint is satisfied, but is not strictly a navigation function.

C. Line-of-Sight

If two robots q_i, q_j such that $e_{ij} \in E$, are in danger of loosing line-of-sight, it means one of them is occluded

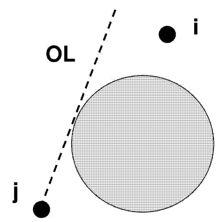

Fig. 4. Geometry of the line of sight problem, required to compute ϕ_i^{los}.

from the other's view by an obstacle as seen in Fig. 4. The line connecting them at the last time when line of sight was satisfied is referred to as the *occlusion line*, OL (see Fig. 4). The line of sight constraint is enforced by a potential:

$$\phi_{ij}^{los}(q_i, q_j) = \begin{cases} 0 & if\ \ L.O.S. \\ d^2(q_i, OL) & else \end{cases} \tag{IV.2}$$

where $d(q_j, OL)$ denotes the distance from q_i to the occlusion line defined in the usual way. Note the potential only possesses minima at configurations where the line-of-sight constraint is satisfied, but is not a proper Navigation Function.

V. Parallel composition

At any time, q_i must select a direction to move, which goes toward its goal position (ideally, $\dot{q}_i = -\frac{\partial}{\partial q_i}\phi_i^{goal}(q_i)$), and for each corresponding edge e_{ij} in G^*, it must maintain range (ideally $\dot{q}_i = -\frac{\partial}{\partial q_i}\phi_{ij}^{range}(q_i, q_j)$) and line of sight (ideally $\dot{q}_i = -\frac{\partial}{\partial q_i}\phi_{ij}^{los}(q_i, q_j)$). However, it is not strictly necessary to use these traditional choices of motion directions.

A. Theory

Proposition 5.1: Any control law u (even a discontinuous one) which satisfies

$$\frac{\partial}{\partial q}\phi(q,t) \cdot u \leq -\frac{\partial \phi(q,t)}{\partial t} \tag{V.1}$$

is acceptable and will stabilize the system to the minima of ϕ [3].

A potential function, ϕ is essentially a Lyapunov function. Provided $\dot{\phi}(q(t)) \leq 0$, $\forall t$ we can show q approaches the minima of the potential function, since a selection of u in accordance with the proposition produces:

$$\dot{\phi} = \frac{\partial}{\partial q}\phi(q,t) \cdot u + \frac{\partial \phi(q,t)}{\partial t} \leq 0. \tag{V.2}$$

Said another way, all control laws that satisfy eq. V.1 create vector fields which possess a Common Lyapunov Function, ϕ.

In the case of $\phi_i^{goal}(q_i)$, there is no explicit time dependence so the partial with respect to time vanishes. In addition, it is uniformly maximal on the boundary of C_{free}, so its obstacle avoidance properties are preserved by all such u [3]. For functions such as $\phi^{los}(q_i, q_j)$ or $\phi^{range}(q_i, q_j)$ there are two ways to apply this. First one could consider ϕ to have no explicit time dependence, but robots i and j are actively working in concert to achieve ϕ by each selecting u_i and u_j to decrease ϕ. In that case select

$$\frac{\partial}{\partial q_i}\phi(q_i, q_j) \cdot u_i \leq 0 \tag{V.3}$$

$$\frac{\partial}{\partial q_j}\phi(q_i, q_j) \cdot u_j \leq 0. \tag{V.4}$$

Alternatively, if only one of the robots is actively working to reduce ϕ then the second robot's influence can be viewed as an explicit time dependence

$$\dot{\phi} = \frac{\partial}{\partial q_i}\phi(q_i, t) \cdot u_i + \frac{\partial \phi(q_i, t)}{\partial t} \leq 0. \tag{V.5}$$

Where the partial with respect to t can be determined if q_j's velocity is known. This information can be sent to robot i via the wireless link and ϕ can still be satisfied.

If one desires to *compose in parallel* (*i.e.*, simultaneously satisfy) multiple objectives encoded by $\phi_1, \ldots, \phi_P$, u_i must satisfy

$$\frac{\partial}{\partial q_i}\phi_l(q_i, t) \cdot u_i \leq -\frac{\partial \phi_l(q_i, t)}{\partial t} \qquad \forall l \in [1, P]$$

Such a u_i is called a *feasible direction*. It is possible to have multiple or no feasible directions.

Remark 5.2: In the case of two objectives $\phi_1(q)$ and $\phi_2(q)$, with no explicit time dependence, there always exists a feasible direction.

B. Computation

In [3] an iterative nonlinear programming method was introduced to compute u which was expensive but worked well in arbitrary dimension. Here we streamline the algorithm for use in $\mathbb{R}^2$ and guarantee that it terminates in fixed number of steps. The algorithm is illustrated in Algorithm 1 and in Figure 5. Each input vector, $v_1, \ldots, v_P$, corresponds to the negative gradients of any relevant potentials robot i must consider. Typically for the problems considered here each robot will have one potential for the goal, and up to two additional potentials (Range and Line-of-Sight) per desired communication link (incident edge in G^*). Note that α is a arbitrary weighting factor that can be selected off-line. We use $\alpha = 0.5$ here. The notation *Rot* means to rotate a vector about an axis by an angle.

There are $P(P-1)/2$ possible cross products to compute. In $\mathbb{R}^2$ each requires 3 floating point operations. Therefore, in the worst case $3P(P-1)/2$ operations are required to test feasibility. If the problem is feasible an additional 16 floating point operations result in an answer for u_i. If the set of vectors has no feasible solution then some directions must be dropped until a feasible solution exists. A prioritization scheme can be used for this purpose, however in our experience this is generally not necessary for the class of problems we consider, as discussed in the following section.

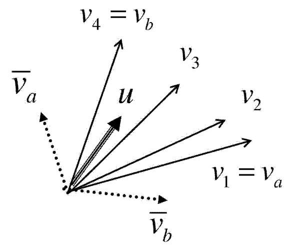

Fig. 5. Given a series of possible input directions $v_1, \ldots, v_P$, the algorithm synthesizes a movement direction u which is in the same half plane as all input vectors.

Algorithm 1 Algorithm for generating a feasible motion direction in $\mathbb{R}^2$

Function u_i = Feasible Direction$(v_1, v_2, \ldots, v_P)$
if $\exists v_a | v_a \times v_j \cdot \hat{k} \geq 0, \forall j \neq a$ **then**
  Find $v_b | v_b \times v_l \cdot \hat{k} \leq 0, \forall l \neq b$
  $\bar{v}_a = Rot(\hat{k}, 90^o) \times v_a$
  $\bar{v}_b = Rot(\hat{k}, -90^o) \times v_b$
  $u_i = \alpha \bar{v}_a + (1-\alpha)\bar{v}_b$, for some $\alpha \in [0, 1]$
else
  Infeasible; END
end if

C. Problem Structure and completeness

Given an edge $e_{ij} \in G^*$, and ϕ_i^{goal} and ϕ_j^{goal}, a feasible direction always exits as long as the Range and Line-of-Sight constraints are not simultaneously active. This follows from the fact that given any two directions, a feasible direction always exists (Remark 5.2).

Guaranteeing the existence of a feasible direction when both Range and Line-of-Sight constraints are active is more difficult. The problem considered here has several special features worth noting.

Remark 5.3: The problem is planar.

Remark 5.4: If $\frac{\partial}{\partial q_i}\phi_i^{range}$ and $\frac{\partial}{\partial q_i}\phi_i^{los}$ are both nonzero, they are perpendicular to one another so their composition always has a feasible direction.

Remark 5.5: For two robots, i and j corresponding to the edge e_{ij}, $\frac{\partial}{\partial q_i}\phi_{ij}^{range} = -\frac{\partial}{\partial q_j}\phi_{ji}^{range}$ (anti-symmetric) and $\frac{\partial}{\partial q_i}\phi_{ij}^{los} = \frac{\partial}{\partial q_j}\phi_{ji}^{los}$ (symmetric).

In light of these observations and the discussion in Sect. V-A, consider Figure 6. Given an edge $e_{ij} \in G^*$, provided $-\nabla\phi_i^{goal}$ *and* $-\nabla\phi_j^{goal}$ do not *both* lie in the region labelled infeasible, there always exists u_i and/or u_j which can simultaneously satisfy each robot's goal objectives as well as the Range and Line-of-Sight constraint for the pair. This can be verified graphically since for any goal direction outside the infeasible region, the goal, line-of-sight and range directions for a given robot all lie in the same half plane. And as long as one robot has a feasible direction, and knows the other robot's velocity the edge can be preserved. The question

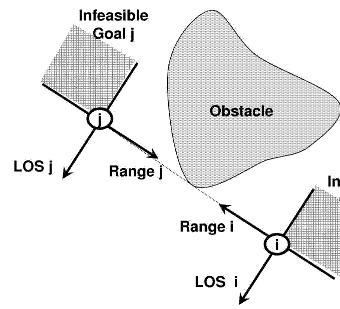

Fig. 6. A generic pair of robots, with active Line-of-Sight and Range constraints. Figure indicates possible directions of $-\frac{\partial}{\partial q}\phi^{goal}$ that could result in no feasible direction for simultaneously satisfying three objectives.

of global convergence remains: Is there any guarantee that both robot's goal vectors will not simultaneously lie in the infeasible region?

Conjecture: Given any $e_{ij} \in G^*$, provided: 1. the lines connecting q_i^o and q_j^o, or q_i^f and q_j^f do not intersect a stable manifold discussed in Remark 4.2; and 2. obstacles are convex; a feasible direction always exists. And, following the motion directions assigned by Algorithm 1, the link e_{ij} is preserved for all time and both robots i and j reach their goals.

Sketch of proof: If both robots are selecting infeasible directions, it means they tend to separated by more than $\rho_{\max}$. Since the goal positions are separated by $\rho_{\max}$, the separation is not attributed to the difference in the fields. The separation behavior implies the points q_i^o and q_j^o intersects a stable manifold. However, at this time a formal proof of this conjecture is the subject of on-going research.

VI. Experiments

MATLAB simulations were used to demonstrate the algorithms operation. Figure 7 shows a swarm of 5 robots (red circles) maintaining the five initial edges shown in frame 1 as they move toward the goal configuration. Green lines indicate a wireless link. Figure 8 depicts the operation of the algorithm with 12 robots and 15 desired edges.

The above algorithm was implemented on Koala robots, made by K-Team, with an on-board PC104 stack. The robots were color coded and an overhead camera was used to provide position information over a wireless network. Each robot is also equipped with a radio modem to allow peer-to-peer wireless communication. Figure 9 shows three snap shots of a scenario. As the robots move toward their goals according to ϕ^{goal}, the range constraint becomes active as seen in the upper snapshot. As the red robot negotiates the obstacle (green), the line-of-sight constraint between the red and orange robots becomes active between the middle and lower frames. Finally in the lower frame, with range and line of sight satisfied, the robots proceed toward their respective goals (defined relative to the small yellow square). The control law easily achieved

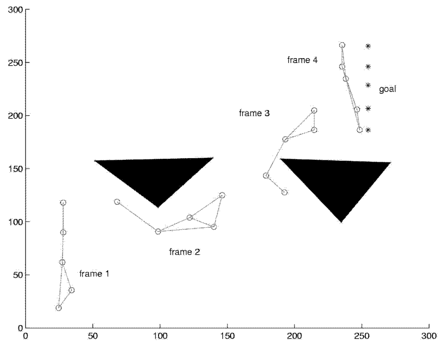

Fig. 7. A swarm of 5 robots maintaining a specified communication graph.

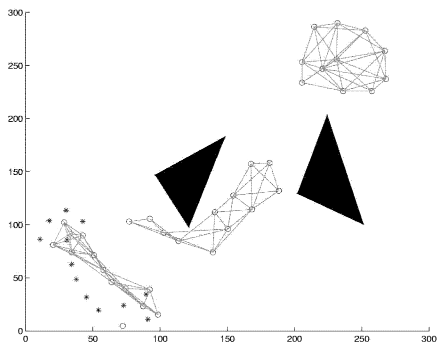

Fig. 8. A swarm of 12 robots maintaining a specified communication graph.

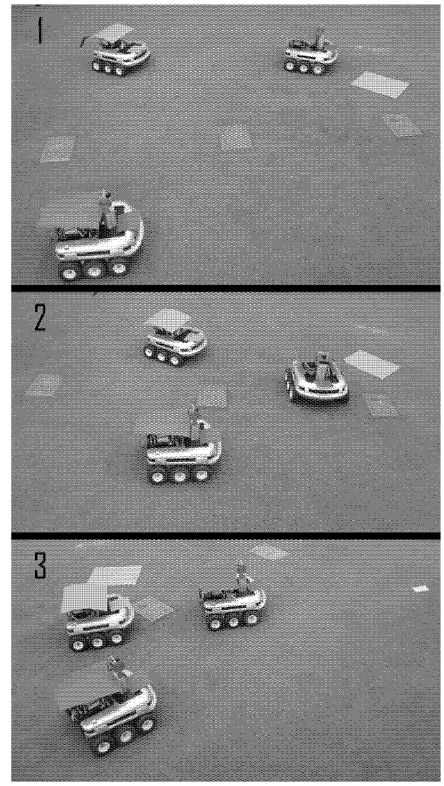

Fig. 9. Three still frames from an experiment showing the robots coming together for communication, avoiding obstacles (green square) and moving towards the goal location (yellow square in frame 3).

"realtime" update rates (>> 100 Hz), with the vision system serving as the limiting factor (10 Hz).

VII. Conclusion

Motivated by the use of wireless communication among swarm members, in this paper we consider the problem of steering N robots to N goals, while maintaining some range and line of sight constraints between them in the presence of obstacles. Range and line of sight are two conditions which improve the reliability of wireless transmission. To the author's knowledge this is the first work to consider the effect of line of sight constraints for swarms. After establishing some basic conditions on the existence of solutions, we show that one rather profound condition is that all robots must pass on the same side of an obstacle (same path-class) for the swarm to remain connected. An implication of this is that, in order to remain connected, the swarm must either have a leader or some on-line method for achieving consensus on the path class. A further consequence of this is that navigation functions do not offer a global solution to this problem because the existence of saddle points makes it impossible to guarantee all robots select the same path class for arbitrary initial conditions. Basic potentials for Range and Line of Sight Constraints were introduced and a method for composing multiple potential functions into a single feasible motion direction was presented. An efficient computational algorithm to compute this direction is proposed. Simulations and hardware based demonstrations of the algorithm's operation are provided and show promising results.

Acknowledgments

The authors were supported by ONR and the Trident Scholar Program.

References

[1] J.P. Desai, J.P.Ostrowski, and V.Kumar. Modeling and control of formations of nonholonomic robots. *IEEE Transactions on Robotics and Automation*, 17(6):905–908, 2001.

[2] D.Spanos and R.Murray. Motion planning with wireless network constraints. In *American Control Conference*, 2005.

[3] J.M. Esposito and V. Kumar. A method for modifying closed loop motion plans to satisfy unpredictable dynamics constraints at run time. In *IEEE International Conference on Robotics and Automation*, pages 1691–1696, Washington, DC, May 2002.

[4] H.G.Tanner, A. Jadbabaie, and G.J.Pappas. Stable flocking of mobile agents, part i: Fixed topology. In *IEEE Conference on Decision and Control*, 2003.

[5] H.G.Tanner, A. Jadbabaie, and G.J.Pappas. Stable flocking of mobile agents, part ii: Dynamic topology. In *IEEE Conference on Decision and Control*, 2003.

[6] J.R. Munkres. *Topology*. Prentice Hall, Englewood Cliffs, NJ, 1975.

[7] E. Rimon and D. E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, 1992.

[8] R.Olfati-Saber and R.Murray. Flocking with obstacle avoidance: cooperation with limited communication in mobile networks. In *IEEE Conference on Decision and Control*, 2003.

[9] V.Gazi and K.M.Passiano. Stability analysis of swarms. *IEEE Transactions on Automatic Control*, 48(4):692–696, 2003.